# sql과 dplyr, tidyr

박찬엽

2017년 6월 27일

# 목차

# 과제 확인

# sql

# CRUD

create: 데이터의 생성
read : 데이터의 조회
update: 데이터의 변경
delete: 데이터의 제거

# 분석에 사용되는 READ

SELECT
FROM
(WHERE)
(GROUP BY)
(ORDER BY)

# select

select는 column을 선택하는 구문입니다. 기존에 있는 데이터를 기준으로 어떤 column을 선택할꺼냐를 묻는 곳으로 전체가 필요하면 *를 사용합니다.

# from

from은 table을 선택하는 구문입니다. 생각 같아서는 어디 테이블에 어디 컬럼이 좋을 것 같은데, 영문권에서는 이 순서가 더 자연스러운 모양입니다.

# where

where는 조건문입니다. 위에 두 경우(table, column)에 해당하는 데이터 전체를 가져오고 싶으면 select * from bank 같이 bank테이블 전체를 가져오는 query를 작성하시면 됩니다. 하지만 그 와중에 데이터들이 조건에 해당하는 일정 부분만 필요하면 그 부분을 where 뒤에 작성하시면 됩니다.

# group by

group by는 선택한 column으로 묶어서 처리하라는 뜻입니다. 예를 들어 A반의 학생들 평균을 알고 싶으면 A반 학생이라는 조건인 성적만 가져와서 평균을 낼 수도 있지만, 반끼리 데이터를 사용하라고 알려주고 평균을 구할 수도 있습니다.

# order by

order by는 정렬을 위한 column을 정할 때 사용합니다. db는 table의 데이터를 저장할 때 순서를 고려해서 저장하지 않습니다. 그렇기 때문에 사람의 편의를 위해 순서를 강제하는 방법을 알려주는 것입니다.

# as

하나 더 as는 길게 만들어진 무엇을 줄여서 쉽게 작성하기 위해서 사용합니다. column 명이나 table명에서 많이 사용합니다.

# data.frame과 비교 1

```
# for R
train

# for db
select * from train
```

# data.frame과 비교 2

```
# for R
train[,c("date","age")]

# for db
select date, age from train
```

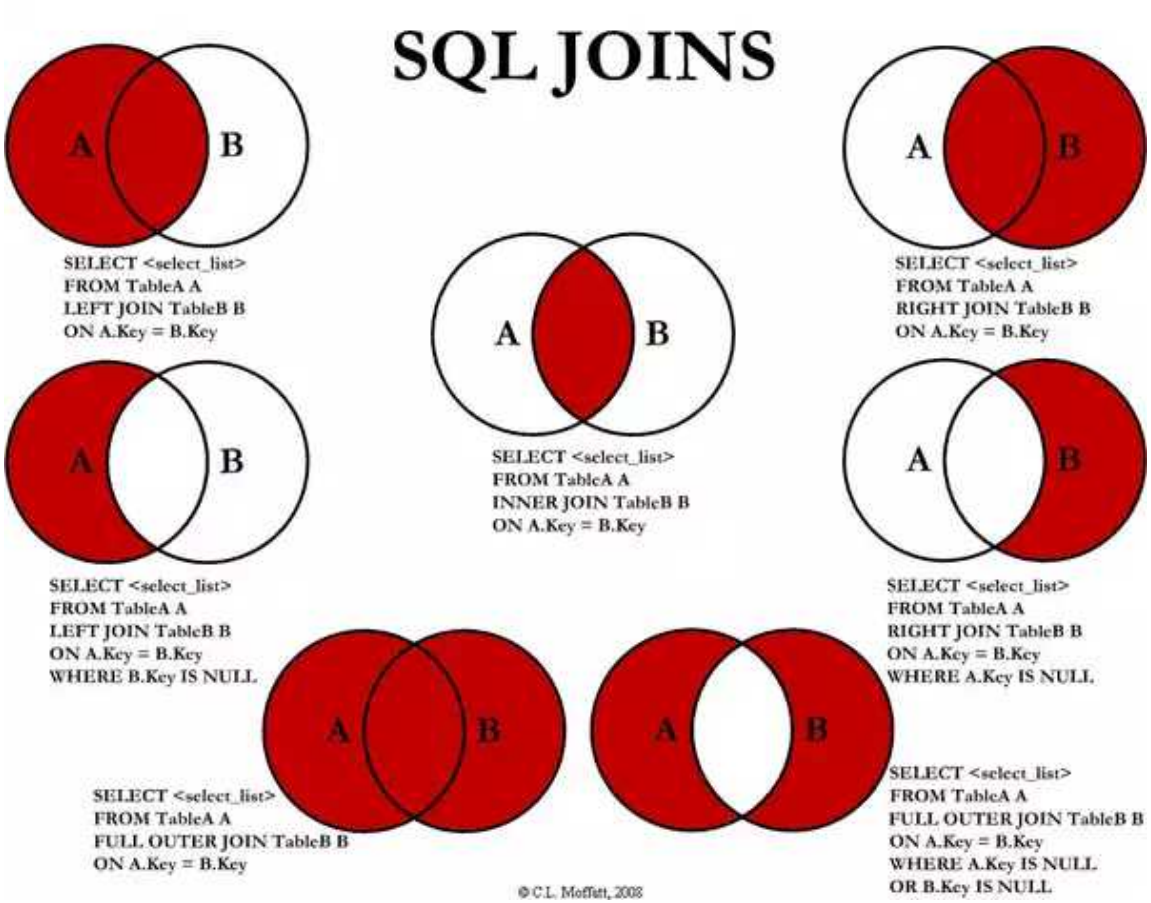
# data.frame과 비교 3

```
# for R
train[train$date > 2016-06-28 ,c("date","age")]

# for db
select date, age from train where train.date > 2016-07-28
```

# data.frame과 비교 4

```
# for R
data<-train[,c("sex","age")]
mean(data[data$sex=="F","age"])
mean(data[data$sex=="M","age"])

# for db
select sex, avg(age) from train group by sex
```

# join


SQL JOINS
A
B
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
© C.L. Moffatt, 2008

# inner join

```
select *
from A inner join B on A.key = b.key;
```

# outer join

```
select *
from A outer join B on A.key = b.key;
```

# left join

```
select *
from A left join B on A.key = b.key;
```

# dplyr + tidyr

# pipe 연산자 %>%

```
g(f(y)) == y %>% f() %>% g()
```

```
g(f(x,y,z)) == y %>% f(x, . , z) %>% g
```

# dplyr

filter : 행에 조건을 줘서 부분을 불러옴
select : 필요한 컬럼만 선택
mutate : 새로운 컬럼을 계산해서 만듬
arrange : 조건에 따라 재정렬
group_by : 그룹을 조건으로 사용
summarise : 요약형 계산을 진행

# tidyr

spread : long form > wide form
gather : wide form > long form
seperate : 한 컬럼내의 데이터를 지정 조건으로 분리
unite : 여러 컬럼의 데이터를 한 컬럼으로 함침
extract : 데이터를 분리하는 폼을 지정하여 분리

# 실습

https://github.com/mrchypark/dabrp_classnote2/blob/master/codeForClass3.R

# data.table

cheat sheet